STEEPEST DESCENT AND CONJUGATE GRADIENT METHODS WITH VARIABLE PRECONDITIONING [*]

ANDREW V. KNYAZEV[†] AND ILYA LASHUK[‡]

Abstract. We analyze the conjugate gradient (CG) method with variable preconditioning for solving a linear system with a real symmetric positive definite (SPD) matrix of coefficients A. We assume that the preconditioner is SPD on each step, and that the condition number of the preconditioned system matrix is bounded above by a constant independent of the step number. We show that the CG method with variable preconditioning under this assumption may not give improvement, compared to the steepest descent (SD) method. We describe the basic theory of CG methods with variable preconditioning with the emphasis on "worst case" scenarios, and provide complete proofs of all facts not available in the literature. We give a new elegant geometric proof of the SD convergence rate bound. Our numerical experiments, comparing the preconditioned SD and CG methods, not only support and illustrate our theoretical findings, but also reveal two surprising and potentially practically important effects. First, we analyze variable preconditioning in the form of inner-outer iterations. In previous such tests, the unpreconditioned CG inner iterations are applied to an artificial system with some fixed preconditioner as a matrix of coefficients. We test a different scenario, where the unpreconditioned CG inner iterations solve linear systems with the original system matrix A. We demonstrate that the CG-SD inner-outer iterations perform as well as the CG-CG inner-outer iterations in these tests. Second, we compare the CG methods using a two-grid preconditioning with fixed and randomly chosen coarse grids, and observe that the fixed preconditioner method is twice as slow as the method with random preconditioning.

Key words. Steepest descent, conjugate gradient, iterative method, inner-outer iterations, variable preconditioning, random preconditioning, preconditioner, condition number, linear systems, circular cone, Householder reflection, convergence rate bound, multigrid.

AMS subject classification. 65F10

1. Introduction. Preconditioning, a transformation, usually implicit, of the original linear system aiming at accelerating the convergence of the approximations to the solution, is typically a necessary part of an efficient iterative technique. Modern preconditioning, e.g., based on so-called algebraic multilevel and domain decomposition methods, attempts to become as close to a "black box" ideal of direct solvers as possible. In this attempt, the mathematical structure of the preconditioner, which in the classical case is regarded as some linear transformation, may become very complex, in particular, the linearity can be easily lost, e.g., if the preconditioning itself involves "inner" iterative solvers. The fact that the preconditioner may be nonlinear, or variable, i.e., changing from iteration to iteration, may drastically affect the known theory as well as the practical behavior of preconditioned iterative methods and therefore needs special attention. Our main result is that the conjugate gradient (CG) method with variable preconditioning in certain situations may not give improvement, compared to the steepest descent (SD) method for solving a linear system with a real symmetric positive definite (SPD) matrix of coefficients. We assume

[*] Compiled February 2, 2008. Received by the editors November 17, 2006; accepted for publication (in revised form) ????????????; published electronically ??????????. Preliminary version of this paper is available as a technical report [9, v1-v3].

[†] Department of Mathematical Sciences University of Colorado at Denver and Health Sciences Center, P.O. Box 173364, Campus Box 170, Denver, CO 80217-3364 (Andrew.Knyazev[at]cudenver.edu, http://math.cudenver.edu/~aknyazev/). This material is based upon work supported by the National Science Foundation awards DMS 0208773 and 0612751

[‡] Department of Mathematical Sciences University of Colorado at Denver and Health Sciences Center, P.O. Box 173364, Campus Box 170, Denver, CO 80217-3364 (Ilya.Lashuk[at]cudenver.edu).

that the preconditioner is SPD on each step, and that the condition number of the preconditioned system matrix is bounded above by a constant.

Let us now introduce the notation, so that we can formulate the main result mathematically. Let A be a real SPD matrix, (x, y) be the standard inner product of real vectors x and y, so that $(Ax, y) = (x, Ay)$, and let $\|x\| = \sqrt{(x, x)}$ be the corresponding vector norm. We also use $\|\cdot\|$ to denote the operator norm. The A-inner product and the A-norm are denoted by $(x, y)_A = (x, Ay)$ and $\|x\|_A = \sqrt{(x, x)_A}$.

We consider a family of iterative methods to obtain a sequence of approximate solutions x_k of a linear system $Ax = b$ and use the A-norm to measure the error $e_k = x - x_k$. The SD and CG methods are well-known iterative procedures that fit into our framework. To accelerate the convergence of the error e_k to zero we introduce preconditioning, i.e., on every iteration k an operator B_k, called the preconditioner, possibly different for each iteration k, is applied to the residual $r_k = b - Ax_k$. A general algorithm, which includes the preconditioned SD or CG (PSD or PCG respectively) methods as particular cases, can be presented as follows, e.g., Axelsson [3, p. 540] and Axelsson and Vassilevski [1, Algorithm 5.3]: given A, b, $\{B_k\}$, $\{m_k\}$, x_0, for $k = 0, 1, \ldots$: $r_k = b - Ax_k$, $s_k = B_k^{-1} r_k$, and

$$p_k = s_k - \sum_{l=k-m_k}^{k-1} \frac{(As_k, p_l)}{(Ap_l, p_l)} p_l, \quad x_{k+1} = x_k + \frac{(r_k, p_k)}{(Ap_k, p_k)} p_k, \tag{1.1}$$

where

$$0 \leq m_k \leq k \text{ and } m_{k+1} \leq m_k + 1. \tag{1.2}$$

The latter condition is highlighted in Notay [11, p. 1447, line 1] and ensures that the formula for p_k in (1.1) performs the standard Gram–Schmidt A-orthogonalizations to previous search directions, which are already pairwise A-orthogonal. The full orthogonalization that performs explicit A-orthogonalizations to all previous search directions corresponds to $m_k = k$. Choosing $m_k = \min\{k, 1\}$ gives the PCG method, e.g., described in Golub and Ye [6, IPCG Algorithm]. The connection of this PCG method to the commonly used PCG algorithm is discussed in section 7 following Golub and Ye [6, Remark 2.3]. The shortest recursion $m_k = 0$ leads to the standard PSD method.

It is well-known, e.g., D′yakonov [4, p. 34] and Axelsson [3, Sec. 11.1.2, p. 458] that if the preconditioner is SPD and fixed, $B_k = B = B^* > 0$, a preconditioned method, such as (1.1), using the preconditioner B can be viewed as the corresponding unpreconditioned method applied to the preconditioned system $B^{-1}Ax = B^{-1}b$ in the B-based inner product $(x, y)_B = (x, By)$. This implies that the theory obtained for unpreconditioned methods remains valid for preconditioned methods, in particular, the A-orthogonalization terms with $l < k-1$ in the sum in (1.1) vanish in exact arithmetic, e.g., Axelsson [3, Sec. 11.2.6, Th. 11.5]. The situation changes dramatically, however, if different preconditioners B_k are used in the PCG method.

The present paper concerns the behavior of method (1.1), where the preconditioner B_k varies from step to step, but remains SPD on each step and the spectral condition number $\kappa\left(B_k^{-1}A\right) = \lambda_{\max}\left(B_k^{-1}A\right)/\lambda_{\min}\left(B_k^{-1}A\right)$ is bounded above by some constant $\kappa_{\max}$ independent of the step number k. We note that the matrix $B_k^{-1}A$ is SPD with respect to, e.g., the B_k inner product, so its eigenvalues are real positive. Let us highlight that our assumption $\kappa\left(B_k^{-1}A\right) \leq \kappa_{\max}$ can be equivalently written as $\|I - B_k^{-1}A\|_{B_k} \leq \gamma$ with $\kappa_{\max} = (1+\gamma)/(1-\gamma)$, assuming without loss of generality that B_k is scaled such that $\lambda_{\max}\left(B_k^{-1}A\right)+\lambda_{\min}\left(B_k^{-1}A\right)=2$. Here, we only deal with methods that are invariant with respect to scaling of B_k.

The main result of this paper is that the preconditioned method (1.1) with (1.2) turns into the PSD method with the worst possible convergence rate on every iteration, if the preconditioners B_k satisfying our assumption $\kappa\left(B_k^{-1}A\right)\leq\kappa_{\max}$ are chosen in a special way. We explicitly construct a variable preconditioner that slows down the CG method to the point that the worst linear convergence rate of the SD method is recovered. Thus one can only guarantee that the convergence rate for the method (1.1) with (1.2) is just the same as for the PSD method, $m_k=0$, obtained in Kantorovič [7] and reproduced, e.g., in Kantorovich and Akilov [8, Ch. XV]:

$$\frac{\|e_{k+1}\|_A}{\|e_k\|_A}\leq\frac{\kappa_{\max}-1}{\kappa_{\max}+1}. \tag{1.3}$$

Our proof is geometric and is based on the simple fact, proved in section 2, that a nonzero vector multiplied by all SPD matrices with a condition number bounded by a constant generates a pointed circular cone. We apply this fact on every iteration to the current residual vector, which becomes the center of the cone, so all points in the cone correspond to all possible preconditioned residuals. In a somewhat similar way, Golub and Ye [6] use the angle between the exact and the perturbed preconditioned residuals. In the CG method context, this cone has a nontrivial intersection with the subspace A-orthogonal to all previous search directions. So on each iteration we can choose a preconditioner with the a priori chosen quality, determined by $\kappa_{\max}$, that makes enforcing A-orthogonality with respect to all previous search directions useless.

Basic properties of method (1.1), most importantly the local optimality, are derived in section 3. In section 4 we apply our results from section 2 about the cone to obtain a new proof of estimate (1.3). In section 5 we analyze the convergence of the PCG method with variable preconditioning and prove our main result. We assume real arithmetic everywhere in the paper, except for section 6, where we show that our main results also hold for complex Hermitian positive definite matrices. In section 7 we consider two particular PCG algorithms that are often used in practice and describe their behavior with variable preconditioning.

Our numerical experiments in section 8 comparing the preconditioned SD and CG methods support and illustrate our theoretical findings, and also reveal some potentially practically important effects. In subsection 8.1, we test the widely used modification of the CG method with a simplified formula for the scalar β_k from section 7 and demonstrate that variable preconditioning can make this modification much slower than even the SD method. In subsection 8.2, we analyze inner-outer iterations as variable preconditioning. Finally, in subsection 8.3, we demonstrate that variable preconditioning may surprisingly accelerate the SD and the CG compared to the use of fixed preconditioning in the same methods.

Different aspects of variable preconditioning are considered, e.g., in Axelsson and Vassilevski [1, 2], Axelsson [3], where rather general nonlinear preconditioning is introduced, and in Golub and Ye [6], Notay [11] that mainly deal with the case when the preconditioner on each iteration approximates a fixed operator. In Axelsson and Vassilevski [2], Axelsson [3], Golub and Ye [6], Notay [11], convergence estimates for some iterative methods with variable preconditioning are proved. For recent results and other aspects of variable preconditioning see Simoncini and Szyld [12, 13, 14] and references there. No attempts are apparently made in the literature to obtain a result similar to ours, even though it should appear quite natural and somewhat expected to experts in the area, after reading this paper.

2. Pointed circular cones represent sets of SPD matrices with varying condition numbers.

For a pair of real non-zero vectors x and y we define the angle between x and y in the usual way as

$$\angle(x,y) = \arccos\left(\frac{(x,y)}{\|x\|\,\|y\|}\right) \in [0,\pi].$$

The following theorem is inspired by Neymeyr [10, Lemma 2.3].

Theorem 2.1. *The set $\{Cx\}$, where x is a fixed nonzero real vector and C runs through all SPD matrices with condition number $\kappa(C)$ bounded above by some $\kappa_{\max}$, is a pointed circular cone, specifically,*

$$\{Cx: \quad C = C^* > 0,\ \kappa(C) \le \kappa_{\max}\} = \left\{y: \quad \sin\angle(x,y) \le \frac{\kappa_{\max}-1}{\kappa_{\max}+1}\right\}.$$

Theorem 2.1 can be proved by constructing our cone as the smallest pointed cone that includes the ball considered in Neymeyr [10, Lemma 2.3]. Preparing for section 6 that deals with the complex case, not covered in Neymeyr [10], we provide a direct proof here based on the following two lemmas. The first lemma is simple and states that the set in question cannot be larger than the cone:

Lemma 2.2. *Let x be a non-zero real vector, let C be an SPD matrix with spectral condition number $\kappa(C)$. Then $\sin\angle(x, Cx) \le (\kappa(C)-1)/(\kappa(C)+1)$.*

Proof. Denote $y = Cx$. We have $(x, Cx) = \left(y, C^{-1}y\right) > 0$ since C is SPD, so $y \neq 0$ and $\angle(x,y) < \pi/2$. A positive scaling of C and thus of y is obviously irrelevant, so let us choose y to be the orthogonal projection of x onto the 1-dimensional subspace spanned by the original y. Then from elementary 2D geometry it follows that $\|y-x\| = \|x\|\sin\angle(x,y)$. The orthogonal projection of a vector onto a subspace is the best approximation to the vector from the subspace, thus

$$\|x\|\sin\angle(x,y) = \|y-x\| \le \|sy-x\| = \|sCx-x\| \le \|sC-I\|\,\|x\|$$

for any scalar s, where I is the identity. Taking $s = 2/\left(\lambda_{\max}(C)+\lambda_{\min}(C)\right)$, where $\lambda_{\min}(C)$ and $\lambda_{\max}(C)$ are the minimal and maximal eigenvalues of C, respectively, we get $\|sC-I\| = (\kappa(C)-1)/(\kappa(C)+1)$. ▯

The second lemma implies that every point in the cone can be represented as Cx for some SPD matrix C with $\kappa(C)$ determined by the opening angle of the cone.

Lemma 2.3. *Let x and y be non-zero real vectors, such that $\angle(x,y) \in \left[0, \frac{\pi}{2}\right)$. Then there exists an SPD matrix C, such that $Cx = y$ and*

$$\frac{\kappa(C)-1}{\kappa(C)+1} = \sin\angle(x,y).$$

Proof. Denote $\alpha = \angle(x,y)$. A positive scaling of vector y is irrelevant, so as in the previous proof we choose y to be the orthogonal projection of x onto the 1-dimensional subspace spanned by the original y, then $\|y-x\| = (\sin\alpha)\|x\|$, so the vectors $y-x$ and $(\sin\alpha)x$ are of the same length. This implies that there exists a Householder reflection H such that $H((\sin\alpha)x) = y-x$, cf. Neymeyr [10, Lemma 2.3], so $(I + (\sin\alpha)H)x = y$. We define $C = I + (\sin\alpha)H$ to get $Cx = y$. Any Householder reflection is symmetric and has only two distinct eigenvalues ± 1, so C is also symmetric and has only two distinct positive eigenvalues $1 \pm \sin\alpha$, as $\alpha \in [0, \pi/2)$, and we conclude that $C > 0$ and $\kappa(C) = (1+\sin\alpha)/(1-\sin\alpha)$. ▯

3. Local optimality of the method with variable preconditioning. Here we discuss some basic properties of method (1.1) with (1.2). We derive a simple, but very useful, error propagation identity in Lemma 3.1. We prove in Lemma 3.2 that the method is well-defined and has a certain local A-orthogonality property, formulated without a proof in Notay [11, formulas (2.1)-(2.2)] and in the important particular case $m_k = \min\{k, 1\}$ proved in Golub and Ye [6, Lemma 2.1]. Using the local A-orthogonality property of Lemma 3.2, we prove the local A-optimality property in Lemma 3.3 by generalizing the result of Golub and Ye [6, Proposition 2.2]. Finally, we derive a trivial Corollary 3.4 from Lemma 3.3, which uses the idea from Golub and Ye [6, p. 1309] of comparison with the PSD method, $m_k = 0$.

The material of this section is inspired by Golub and Ye [6] and may be known to experts in the field, e.g., some even more general facts can be found in Axelsson [3, Sec. 12.3.2, Lemma 12.22]. We provide straightforward and complete proofs here suitable for a general audience.

Lemma 3.1. *Let A and $\{B_k\}$ be SPD matrices. Suppose p_k in method (1.1) is well-defined and nonzero. Then*

$$e_{k+1} = e_k - \frac{(Ae_k, p_k)}{(Ap_k, p_k)} p_k. \tag{3.1}$$

Proof. Recall that $e_k = A^{-1}b - x_k$ and thus $r_k = Ae_k$. Then (3.1) follows immediately from the last formula in (1.1). □

Lemma 3.2. *Let A and $\{B_k\}$ be SPD matrices and $\{m_k\}$ satisfies (1.2). Then the error, the preconditioned residual, and the direction vectors generated by method (1.1) before the exact solution is obtained are well-defined and satisfy*

$$(p_i, p_j)_A = 0, \; k - m_k \le i < j \le k, \tag{3.2}$$

$$(e_{k+1}, s_k)_A = (e_{k+1}, p_i)_A = 0, \; k - m_k \le i \le k. \tag{3.3}$$

Proof. We first notice that (3.1) for any k obviously implies

$$(e_{k+1}, p_k)_A = 0. \tag{3.4}$$

For the rest of the proof we use an induction in k. Let us take $k = 0$ and suppose $x_0 \ne x$, then $r_0 \ne 0$ and $s_0 \ne 0$ since B_0 is SPD. By (1.2), $m_0 = 0$ and thus $p_0 = s_0 \ne 0$, so in the formula for x_{k+1} we do not divide by zero, i.e., x_{k+1} is well defined. There is nothing to prove in (3.2) for $k = 0$ since $m_0 = 0$. Formula (3.4) implies $(e_1, p_0)_A = (e_1, s_0)_A = 0$, i.e., (3.3) holds for $k = 0$. This provides the basis for the induction.

Suppose the statement of the lemma holds for $k - 1$, which is the induction hypothesis, i.e., up to the index $k - 1$ all quantities are well defined and

$$(p_i, p_j)_A = 0, \; k - 1 - m_{k-1} \le i < j \le k - 1, \tag{3.5}$$

$$(e_k, s_{k-1})_A = (e_k, p_i)_A = 0, \; k - 1 - m_{k-1} \le i \le k - 1. \tag{3.6}$$

We now show by contradiction that $x_k \ne x$ implies $p_k \ne 0$. Indeed. if $p_k = 0$ then s_k is a linear combination of $p_{k-m_k}, \ldots, p_{k-1}$. However, since $m_k \le m_{k-1} + 1$, it follows from (3.6) that

$$(e_k, p_i)_A = 0, \; k - m_k \le i \le k - 1. \tag{3.7}$$

Then we have $(s_k, e_k)_A = 0$. At the same time, since the matrix $B_k^{-1}A$ is A-SPD, $s_k = B_k^{-1}Ae_k$ cannot be A-orthogonal to e_k unless $s_k = e_k = 0$, i.e., $x_k = x$.

Next, we prove (3.2) by showing that the formula for p_k in (1.1) is a valid step of the Gram-Schmidt orthogonalization process with respect to the A-based inner product. If $m_k = 0$, there is nothing to prove. If $m_k = 1$ then (3.2) gets reduced to $(p_k, p_{k-1})_A = 0$, which follows from the formula for p_k in (1.1). If $m_k \geq 2$ then condition (1.2) implies that vectors $p_{k-m_k}, \ldots, p_{k-1}$ are among the vectors $p_{k-1-m_{k-1}}, \ldots, p_{k-1}$ and therefore are already A-orthogonal by the induction assumption (3.5). Then the formula for p_k in (1.1) is indeed a valid step of the Gram-Schmidt orthogonalization process with respect to the A-based inner product, so (3.2) holds.

It remains to prove (3.3). We have already established (3.2), and (3.4)–(3.7). Equalities (3.2) and (3.7) imply that p_k and e_k are A-orthogonal to $p_{k-m_k}, \ldots, p_{k-1}$. Equality (3.1) implies that e_{k+1} is a linear combination of e_k and p_k. Thus, we have $(e_{k+1}, p_i)_A = 0,\ k - m_k \leq i \leq k-1$. Finally, it is enough to notice that s_k is a linear combination of $p_k, p_{k-1}, \ldots, p_{k-m_k}$, so $(e_{k+1}, s_k)_A = 0$. ☐

We now use Lemma 3.2 to prove the local optimality of method (1.1) with (1.2), which generalizes the statement of Golub and Ye [6, Proposition 2.2].

Lemma 3.3. *Under the assumptions of Lemma 3.2,*

$$\|e_{k+1}\|_A = \min_{p \in span\left\{s_k, p_{k-m_k}, \ldots, p_{k-1}\right\}} \|e_k - p\|_A.$$

Proof. We get $e_{k+1} \in e_k + \operatorname{span}\{s_k, p_{k-m_k}, \ldots, p_{k-1}\}$ from the formula for p_k in (1.1) and (3.1). Putting this together with A-orthogonality relations (3.3) of the vector e_{k+1} with all vectors that span the subspace finishes the proof. ☐

Two important corollaries follow immediately from Lemma 3.3 by analogy with Golub and Ye [6, Proposition 2.2].

Corollary 3.4. *The A-norm of the error $\|e_{k+1}\|_A$ in method (1.1) with (1.2) is bounded above by the A-norm of the error of one step of the PSD method, $m_k = 0$, using the same x_k as the initial guess and B_k as the preconditioner, i.e., specifically, $\|e_{k+1}\|_A \leq \min_\alpha \|e_k - \alpha s_k\|_A$.*

Corollary 3.5. *Let $m_k > 0$, then the A-norm of the error $\|e_{k+1}\|_A$ in method (1.1) with (1.2) for $k > 0$ satisfies $\|e_{k+1}\|_A \leq \min_{\alpha,\beta} \|e_k - \alpha s_k - \beta(e_k - e_{k-1})\|_A$.*

Proof. Under the lemma assumptions, the formula for p_k in (1.1) and (3.1) imply that $e_{k+1} \in e_k + \operatorname{span}\{s_k, p_{k-1}\} = e_k + \operatorname{span}\{s_k, e_k - e_{k-1}\}$, and the A-orthogonality relations (3.3) turn into $(e_{k+1}, s_k)_A = 0$ and $(e_{k+1}, p_{k-1})_A = (e_{k+1}, e_k - e_{k-1})_A = 0$, so the vector e_{k+1} is A-orthogonal to both vectors that span the subspace. As in the proof of Lemma 3.3, the local A-orthogonality implies the local A-optimality. ☐

Corollary 3.4 allows us in section 4 to estimate the convergence rate of method (1.1) with (1.2) by comparison with the PSD method, $m_k = 0$,—this idea is borrowed from Golub and Ye [6, p. 1309]. The results of Lemma 3.3 and Corollary 3.5 seem to indicate that an improved convergence rate bound of method (1.1) with (1.2) can be obtained, compared to the PSD method convergence rate bound that follows from Corollary 3.4. Our original intent has been to combine Corollary 3.5 with convergence rate bounds of the heavy ball method, in order to attempt to prove such an improved convergence rate bound. However, our results of section 5 demonstrate that this improvement is impossible under our only assumption $\kappa\left(B_k^{-1}A\right) \leq \kappa_{\max}$, since one can construct such preconditioners B_k that, e.g., the minimizing value of β in Corollary 3.5 is zero, so Corollary 3.5 gives no improvement compared to Corollary 3.4.

4. Convergence rate bounds for variable preconditioning. The classical Kantorovich and Akilov [8, Ch. XV] convergence rate bound (1.3) for the PSD method is "local" in the sense that it relates the A-norm of the error on two subsequent iterations and does not depend on previous iterations. Thus, it remains valid when the preconditioner B_k changes from iteration to iteration, while the condition number $\kappa\left(B_k^{-1}A\right)$ is bounded above by some constant $\kappa_{\max}$ independent of k. The goal of this section is to give an apparently new simple proof of the estimate (1.3) for the PSD method, based on our cone Theorem 2.1, and to extend this statement to cover the general method (1.1) with (1.2), using Corollary 3.4.

We denote the angle between two real nonzero vectors with respect to the A-based inner product by

$$\angle_A(x,y) = \arccos\left(\frac{(x,y)_A}{\|x\|_A\,\|y\|_A}\right) \in [0,\pi]$$

and express the error reduction ratio for the PSD method in terms of the angle with respect to the A-based inner product:

Lemma 4.1. *On every step of the PSD algorithm, (1.1) with $m_k=0$, the error reduction factor takes the form $\|e_{k+1}\|_A/\|e_k\|_A = \sin(\angle_A(e_k, B_k^{-1}Ae_k))$.*

Proof. By (3.3), we have $(e_{k+1},p_k)_A = 0$. Now, for $m_k=0$, in addition, $p_k=s_k$, so $0=(e_{k+1},p_k)_A=(e_{k+1},s_k)_A=(e_{k+1},x_{k+1}-x_k)_A$, i.e., the triangle with vertices x, x_k, x_{k+1} is right-angled in the A-inner product, where the hypotenuse is $e_k = x-x_k$. Therefore, $\|e_{k+1}\|_A / \|e_k\|_A = \sin(\angle_A(e_k, x_{k+1}-x_k)) = \sin(\angle_A(e_k,s_k))$, where $s_k = B_k^{-1}(b-Ax_k) = B_k^{-1}Ae_k$ by (1.1). □

Let us highlight that Lemma 4.1 provides an exact expression for the error reduction factor, not just a bound—we need this in the proof of Theorem 5.1 in the next section. Combining the results of Lemmas 2.2 and 4.1 together immediately leads to (1.3) for the PSD method, where $m_k=0$. Finally, taking into account Corollary 3.4, by analogy with the arguments of Golub and Ye [6, p. 1309] and decrypting a hidden statement in Golub and Ye [6, Lemma 3.5], we get

Theorem 4.2. *Convergence rate bound (1.3) holds for method (1.1) with (1.2).*

5. The convergence rate bound is sharp. Here we formulate and prove the main result of the paper that one can only guarantee the convergence rate described by (1.3) for method (1.1) with (1.2) with variable preconditioning if one only assumes $\kappa\left(B_k^{-1}A\right) \le \kappa_{\max}$. Let us remind the reader that (1.3) also describes the convergence rate for the PSD method, (1.1) with $m_k=0$. We now show that adding more vectors to the PSD iterative recurrence results in no improvement in convergence, if a specially constructed set of variable preconditioners is used.

Theorem 5.1. *Let an SPD matrix A, vectors b and x_0, and $\kappa_{\max}>1$ be given. Assuming that the matrix size is larger than the number of iterations, one can choose a sequence of SPD preconditioners B_k, satisfying $\kappa(B_k^{-1}A)\le\kappa_{\max}$, such that method (1.1) with (1.2) turns into the PSD method, (1.1) with $m_k=0$, and on every iteration*

$$\frac{\|e_{k+1}\|_A}{\|e_k\|_A} = \frac{\kappa_{\max}-1}{\kappa_{\max}+1}. \tag{5.1}$$

Proof. We construct the sequence B_k by induction. First, we choose any vector q_0, such that $\sin\angle_A(q_0,e_0) = (\kappa_{\max}-1)/(\kappa_{\max}+1)$. According to Lemma 2.3 applied in the A-inner product, there exists an A-SPD matrix C_0 with condition number $\kappa(C_0)=\kappa_{\max}$, such that $C_0e_0=q_0$. We define the SPD $B_0 = AC_0^{-1}$, then $\kappa(B_0^{-1}A) =$

$\kappa(C_0) = \kappa_{\max}$. We have $s_k = B_k^{-1}Ae_k$, so such a choice of B_0 implies $s_0 = q_0$. Also, we have $p_0 = s_0$, i.e., the first step is always a PSD step, thus, by Lemma 4.1 we have proved (5.1) for $k = 0$. Note that $(e_1, p_0)_A = 0$ by (3.3).

Second, we make the induction assumption: let preconditioners B_l for $l \le k-1$ be constructed, such that $\|e_{l+1}\|_A / \|e_l\|_A = (\kappa_{\max} - 1)/(\kappa_{\max} + 1)$ and $(e_k, p_l)_A = 0$ hold for all $l \le k-1$. The dimension of the space is greater than the total number of iterations by our assumption, so there exists a vector u_k, such that $(u_k, p_l)_A = 0$ for $l \le k-1$ and u_k and e_k are linearly independent. Then the 2D subspace spanned by u_k and e_k is A-orthogonal to p_l for $l \le k-1$.

Let us consider the boundary of the pointed circular cone made of vectors q_k satisfying the condition $\sin\angle_A(q_k, e_k) = (\kappa_{\max} - 1)/(\kappa_{\max} + 1)$. This conical surface has a nontrivial intersection with the 2D subspace spanned by u_k and e_k, since e_k is the cone axis. Let us choose vector q_k in the intersection, This vector will be obviously A-orthogonal to p_l, $l \le k-1$.

Applying the same reasoning as for constructing B_0, we deduce that there exists an SPD B_k such that $\kappa(B_k^{-1}A) \le \kappa_{\max}$ and $B_k^{-1}Ae_k = q_k$. With such a choice of B_k we have $s_k = q_k$. Since $q_k = s_k$ is A-orthogonal to p_l for all $l \le k-1$, it turns out that $p_k = s_k$, no matter how $\{m_k\}$ are chosen. This means that x_{k+1} is obtained from x_k by a steepest descent step. Then we apply Lemma 4.1 and conclude that (5.1) holds. We note, that $(e_{k+1}, p_l)_A = 0$ for all $l \le k$. Indeed, $(e_{k+1}, p_l)_A = 0$ for all $l \le k-1$ since e_{k+1} is a linear combination of e_k and $p_k = s_k = q_k$, both A-orthogonal to p_l for $l \le k-1$. Finally, $(e_{k+1}, p_k)_A = 0$ by (3.3). This completes the construction of $\{B_k\}$ by induction and thus the proof. □

Let us highlight that the statement of Theorem 5.1 consists of two parts: first, it is possible to have the PCG method with variable preconditioning that converges not any faster than the PSD method with the same preconditioning; and second, moreover, it is possible that the PCG method with variable preconditioning converges not any faster than the worst possible theoretical convergence rate for the PSD method described by (1.3). Numerical tests in section 8 show that the former possibility is more likely than the latter. Specifically, we demonstrate numerically in subsection 8.3 that the PCG and PSD methods with random preconditioning converge with the same speed, but both are much faster than what bound (1.3) predicts.

6. Complex Hermitian case. In all other sections of this paper we assume for simplicity that matrices and vectors are real. However, our main results also hold when matrices A and $\{B_k\}$ are complex Hermitian positive definite. In this section we discuss necessary modifications to statements and proofs in sections 2, 4 and 5 in order to cover the complex Hermitian case.

In section 2, the first thing to be changed is the definition of the angle between two non-zero vectors $x, y \in \mathbb{C}^n$, where an absolute value is now taken,

$$\angle(x, y) = \arccos\left|\frac{(x, y)}{\|x\|\,\|y\|}\right| \in [0, \frac{\pi}{2}],$$

that makes the angle acute and invariant with respect to complex nonzero scaling of the vectors. Lemma 2.2 remains valid in the complex case:

Lemma 6.1. *Let x be a non-zero complex vector, and C be a complex Hermitian positive definite matrix with the spectral condition number $\kappa(C)$, then* $\sin\angle(x, Cx) \le (\kappa(C) - 1)/(\kappa(C) + 1)$.

Proof. Denote $y = Cx$ and let $\gamma = (y, x)/\|y\|^2$ then γy is the projection of x onto span$\{y\}$, and $\angle(x, y) = \angle(x, \gamma y)$. Moreover, $(x, \gamma y) = (x, y)\gamma = (x, y)(y, x)/\|y\|^2$ is real—we need this fact later in the proof of Lemma 6.2. We redefine y to γy. The rest of proof is exactly the same as that of Lemma 2.2, since the identity $\|y-x\| = \|x\| \sin\angle(x,y)$, where y is scaled by a complex scalar to be the orthogonal projection of x onto span$\{y\}$, holds in the complex case with the new definition of the angle. ∎

Lemma 2.3 and, thus, Theorem 2.1 do not hold in the complex case after the straightforward reformulation. A trivial counterexample is a pair of vectors $x \neq 0$ and $y = ix$—the angle between x and y is obviously zero, yet it is impossible that $y = Cx$ for any complex Hermitian matrix C, since the inner product $(x,y) = -i\|x\|^2$ is not a real number. This counterexample also gives an idea for a simple fix:

Lemma 6.2. *Let x and y be non-zero complex vectors, such that $\angle(x,y) \neq \pi/2$. Then there exists a complex Hermitian positive definite matrix C and a complex scalar γ, such that $Cx = \gamma y$ and $(\kappa(C)-1)/(\kappa(C)+1) = \sin\angle(x,y)$.*

Proof. We first scale the complex vector y as in the proof of Lemma 6.1 to make y to be the projection of x onto span$\{y\}$. The rest of the proof is similar to that of Lemma 2.3, but we have to be careful working with the Householder reflection in the complex case, so we provide the complete proof.

The redefined y is the projection of x onto span$\{y\}$, thus, $\|y-x\| = (\sin\alpha)\|x\|$, so the vectors $u = y-x$ and $v = (\sin\alpha)x$ are of the same length. Moreover, their inner product (u,v) is real, since (x,y) is real, see the proof of Lemma 6.1. This implies that the Householder reflection $Hz = z - 2(w,z)w$, where $w = (u-v)/\|u-v\|$, acts on $z = u$ such that $Hu = v$, i.e., $H((\sin\alpha)x) = y - x$, so $(I + (\sin\alpha)H)x = y$. We define $C = I + (\sin\alpha)H$ to get $Cx = y$.

The Householder reflection H is Hermitian and has only two distinct eigenvalues ± 1, so C is also Hermitian and has only two distinct positive eigenvalues $1 \pm \sin\alpha$, as $\alpha \in [0,\pi/2)$, and we conclude that $C > 0$ and $\kappa(C) = (1+\sin\alpha)/(1-\sin\alpha)$. ∎

The same change then makes Theorem 2.1 work in the complex case:

Theorem 6.3. *The set $\{\gamma Cx\}$, where x is a fixed nonzero complex vector, γ runs through all nonzero complex scalars, and C runs through all complex Hermitian positive definite matrices with condition number $\kappa(C)$ bounded above by some $\kappa_{\max}$, is a pointed circular cone, specifically,*

$$\{\gamma Cx : \quad \gamma \neq 0, C = C^* > 0,\ \kappa(C) \leq \kappa_{\max}\} = \left\{ y : \quad \sin\angle(x,y) \leq \frac{\kappa_{\max}-1}{\kappa_{\max}+1} \right\}.$$

Section 3 requires no changes other then replacing "SPD" with "Hermitian positive definite." In section 4 we just change the definition of the A-angle to

$$\angle_A(x,y) = \arccos\left| \frac{(x,y)_A}{\|x\|_A \|y\|_A} \right| \in [0, \frac{\pi}{2}],$$

and then Lemma 4.1 holds without any further changes.

Finally, the statement of Theorem 5.1 from section 5 allows for a straightforward generalization:

Theorem 6.4. *Let a Hermitian positive definite matrix A, complex vectors b and x_0, and $\kappa_{\max} > 1$ be given. Assuming that the matrix size is larger than the number of iterations, one can choose a sequence of Hermitian positive definite preconditioners B_k, satisfying $\kappa(B_k^{-1}A) \leq \kappa_{\max}$, such that method (1.1) with (1.2) turns into the PSD method, (1.1) with $m_k = 0$, and on every iteration*

$$\frac{\|e_{k+1}\|_A}{\|e_k\|_A} = \frac{\kappa_{\max}-1}{\kappa_{\max}+1}. \tag{6.1}$$

Proof. Only a small change in the proof of Theorem 5.1 is needed. We first choose any vector q_0', satisfying $\sin \angle_A(q_0, e_0) = (\kappa_{\max} - 1)/(\kappa_{\max} + 1)$. Then by Lemma 6.2 we obtain the complex Hermitian positive definite matrix C_0 and the complex scalar γ such that $C_0 e_0 = \gamma q_0'$. Finally, we choose q_0 to be $\gamma q_0'$ and continue as in the proof of Theorem 5.1. The same modification is made in the choice of the vectors q_k for $k \geq 1$ later in the proof. □

7. Practical PCG algorithms. In this section we briefly discuss two particular well-known PCG algorithms that are often used in practice. Our discussion here is motivated by and follows Golub and Ye [6, Remark 2.3]. Suppose A, b, x_0, $r_0 = b - Ax_0$, $\{B_k\}$ for $k = 0, 1, \ldots$ are given and consider Algorithm 7.1 where β_k on line 7.1 is defined either by expression

$$\beta_k = \frac{(s_k, r_k)}{(s_{k-1}, r_{k-1})}, \tag{7.1}$$

or by expression

$$\beta_k = \frac{(s_k, r_k - r_{k-1})}{(s_{k-1}, r_{k-1})}. \tag{7.2}$$

Formula (7.1) is more often used in practice compared to (7.2), since it can be implemented in such a way that does not require storing the extra vector r_{k-1}.

Algorithm 7.1

```
1:  for $k = 0, 1, \ldots$ do
2:      $s_k = B_k^{-1} r_k$
3:      if $k = 0$ then
4:          $p_0 = s_0$
5:      else
6:          $p_k = s_k + \beta_k p_{k-1}$ (where $\beta_k$ is defined by either (7.2) or (7.1) for all iterations)
7:      end if
8:      $\alpha_k = \frac{(s_k, r_k)}{(p_k, Ap_k)}$
9:      $x_{k+1} = x_k + \alpha_k p_k$
10:     $r_{k+1} = r_k - \alpha_k A p_k$
11: end for
```

If the preconditioner is SPD and fixed, it is well-known, e.g., Golub and Ye [6, Remark 2.3], that $(s_k, r_{k-1}) = 0$, so formula (7.2) coincides with (7.1) and Algorithm 7.1 is described by (1.1) with $m_k = \min(k, 1)$. Of course, in this case the choice $m_k = \min(k, 1)$ is enough to keep *all* search directions A-orthogonal in exact arithmetic.

Things become different when variable preconditioning is used. It is well-known, e.g., Golub and Ye [6, Remark 2.3] and Notay [11, Table 2], that using formula (7.1) for β_k can significantly slow down the convergence, and we provide our own numerical evidence of that in section 8. At the same time, comparing Lemma 3.2 with Lemma 2.1 from Golub and Ye [6], we can show, see Knyazev and Lashuk [9, v1], that Algorithm 7.1 with β_k defined by (7.2), which is exactly Golub and Ye [6, IPCG Algorithm], is equivalent to the particular case of (1.1), namely with $m_k = \min(k, 1)$, and therefore is guaranteed by Theorem 4.2 to converge with at least the same speed as the PSD method.

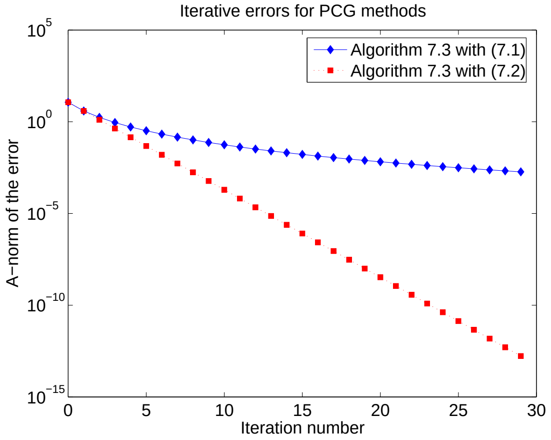

FIG. 8.1. *Algorithm 7.1 with (7.1) fails to provide the PSD convergence rate.*

8. Numerical experiments. We first illustrate the main theoretical results of the paper numerically for a model problem. We numerically investigate the influence of the choice for β_k between formulas (7.1) and (7.2) in Algorithm 7.1 and observe that (7.2) leads to the theoretically predicted convergence rate, while (7.1) may significantly slow down the convergence. Second, we test the convergence of inner-outer iteration schemes, where the inner iterations play the role of the variable preconditioning in the outer PCG iteration, and we illustrate our main conclusion that variable preconditioning may effectively reduce the convergence speed of the PCG method to the speed of the PSD method. Third, and last, we test the PSD and PCG methods with preconditioners of the same quality chosen randomly. We observe a surprising acceleration of the PCG method compared to the use of only one fixed preconditioner; at the same time, we show that the PSD method with random preconditioners works as well as the PCG method, which explains the PCG acceleration and again supports our main conclusion.

8.1. Numerical illustration of the main results. Here, we use the standard 3-point approximation of the 1-D Laplacian of the size 200 as the matrix A of the system. To simulate the application of the variable preconditioner, we essentially repeat the steps described in the proof of Theorem 5.1, i.e., we fix the condition number $\kappa\left(B_k^{-1}A\right) = 2$ and on each iteration we generate a pseudo-random vector s_k, which is A-orthogonal to previous search directions and such that the A-angle between s_k and e_k satisfies $\sin\left(\angle_A\left(s_k, e_k\right)\right) = (\kappa-1)/(\kappa+1)$.

We summarize the numerical results of this subsection on Figure 8.1, where the horizontal axis represents the number of iterations and the vertical axis represents the A-norm of the error. The iteration count actually starts from 1, so the A-norm of the error on the 0-th iteration $\|e_0\|_A$ is just the A-norm of the initial error. The straight dotted (red in the colored print) line marked with squares on Figure 8.1 represents the PSD theoretical bound (1.3) and at the same time it perfectly coincides, which illustrates the statements of Theorem 5.1, with the change of the A-norm of the error in the case where the complete A-orthogonalization is performed, i.e., $m_k = k$ in method (1.1), as well as in the case where Algorithm 7.1 with β_k defined by (7.2) is used. The curved solid (blue) line marked with diamonds represents the convergence

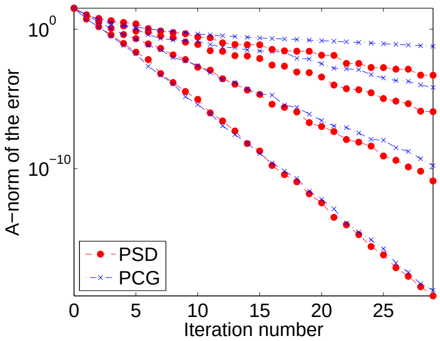

FIG. 8.2. *The PSD and PCG methods with preconditioning by inner CG with different stopping criteria* $\eta = 0.2, 0.4, 0.6$ *and* 0.8 *(from the bottom to the top).*

of Algorithm 7.1 with β_k defined by (7.1), which visibly performs much worse in this test compared to Algorithm 7.1 with (7.2). The paper Notay [11, Sec. 5.2] contains analogous results comparing the change in the convergence rate using formulas (7.1) and (7.2), but it misses a comparison with the PSD method. To check our results of section 6, we repeat the tests in the complex arithmetic. The figure generated is similar to Figure 8.1, so we do not reproduce it here.

8.2. Inner-outer iterations as variable preconditioning. Inner-outer iterative schemes, where the inner iterations play the role of the variable preconditioner in the outer PCG iteration is a traditional example of variable preconditioning; see, e.g., Golub and Ye [6], Notay [11]. Previously published tests analyze an approximation of some fixed preconditioner, $B_k \approx B$, different from A, by inner iterations, typically using the PCG method. The quality of the approximation is determined by the stopping criteria of the inner PCG method. A typical conclusion is that the performance of the outer PCG method improves and it starts behaving like the PCG method with the fixed preconditioner B when B_k approximates B more accurately by performing more inner iterations.

The idea of our tests in this subsection is different: we approximate $B_k \approx B = A$. The specific setup is the following. We take a diagonal matrix A with all integer entries from 1 to 2000, with the right-hand side zero and a random normally distributed zero mean initial guess, and we do the same for the PSD and PCG methods. For preconditioning on the k-th step, applied to the residual r_k, we run the standard CG method without preconditioning as inner iterations, using the zero initial approximation, and for the stopping criteria we compute the norm of the true residual at every inner iteration and iterate until it gets smaller than $\eta\|r_k\|$ for a given constant η. On Figure 8.2, we demonstrate the performance of the PSD and PCG methods for four values of $\eta = 0.2, 0.4, 0.6$ and 0.8 (from the bottom to the top). We observe that the PSD, displayed using dashed (red in the colored print) lines marked with circles and PCG shown as dash-dot (blue) lines with x-marks methods both converge with a similar rate, for each tested value of η. We notice here that the PSD method is even a bit faster than the PCG method. This does not contradict our Corollary 3.4, since the preconditioners B_k here are evidently different in the PSD and PCG methods even though they are constructed using the same principle.

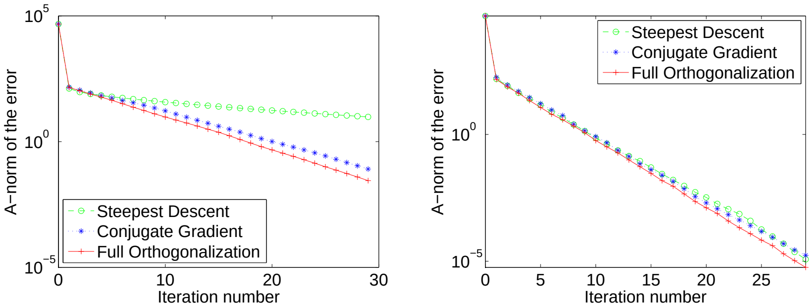

FIG. 8.3. *Two-grid preconditioning with fixed (left) and random (left) coarse grids.*

8.3. Random vs. fixed preconditioning. In this subsection, we numerically investigate a situation where random preconditioners of a similar quality are used in the course of iterations. The system matrix is the standard 3-point finite-difference approximation of the one-dimensional Laplacian using 3000 uniform mesh points and the Dirichlet boundary conditions. We test the simplest multigrid preconditioning using two grids, where the number of coarse grid points is 600. The interpolation is linear, the restriction is the transpose of the interpolation, and the coarse-grid operator is defined by the Galerkin condition. The smoother is the Richardson iteration. On Figure 8.3 left, we once choose (pseudo-)randomly 600 coarse mesh points and build the fixed two-grid preconditioner, based on this choice. On Figure 8.3 right, we choose 600 new random coarse mesh points and rebuild the two-grid preconditioner on each iteration. We note that in the algebraic multigrid the geometric information about the actual position of the coarse grid points is not available, so the random choice of the coarse grids may be an interesting alternative to traditional approaches.

Figure 8.3 displays the convergence history for the PSD (top), PCG (middle), and PCG with the full orthogonalization (bottom) with the same random initial guess using the fixed (left) and variable (right) two-grid preconditioners. On Figure 8.3 left, for a fixed preconditioner, we observe the expected convergence behavior, with the PSD being noticeably the slowest, and the PCG with the full orthogonalization being slightly faster than the standard PCG. Figure 8.3 right demonstrates that all three methods with the variable random preconditioner converge with essentially the same rate, which again illustrates the main result of the paper that the PCG method with variable preconditioning may just converge with the same speed as the PSD method.

Figure 8.3 reveals a surprising fact that the methods with random preconditioning converge twice as fast as the methods with fixed preconditioning! We highlight that Figure 8.3 shows a typical case, not a random outlier, as we confirm by repeating the fixed preconditioner test in the left panel for *every* random preconditioner used in the right panel of Figure 8.3 and by running the tests multiple times with different seeds. Our informal explanation for the fast convergence of the PSD method with random preconditioning is based on Lemma 4.1 that provides the exact expression for the error reduction factor as $\sin(\angle_A(e_k, B_k^{-1}Ae_k))$. It takes its largest value only if e_k is one of specific linear combination of the eigenvectors of $B_k^{-1}Ae$ corresponding to the two extreme eigenvalues. If B_k is fixed, the error e_k in the PSD method after several first iterations approaches these magic linear combinations, e.g., Forsythe [5], and the convergence rate reaches its upper bound. If B_k changes randomly, as in our test, the average "effective" angle is smaller, i.e., the convergence is faster.

Conclusions. We use geometric arguments to investigate the behavior of the PCG methods with variable preconditioning under a rather weak assumption that the quality of the preconditioner is fixed. Our main result is negative in its nature: we show that under this assumption the PCG method with variable preconditioning may converge as slow as the PSD method, moreover, as the PSD method with the slowest rate guaranteed by the classical convergence rate bound. In particular, that gives the negative answer, under our assumption, to the question asked in Golub and Ye [6, Sec. 6, Conclusion.] whether better bounds for the steepest descent reduction factor may exists for Algorithm 7.1 with (7.2).

Stronger assumptions on variable preconditioning, e.g., such as made in Golub and Ye [6], Notay [11] that the variable preconditioners are all small perturbations of some fixed preconditioner, are necessary in order to hope to prove a convergence rate bound of the PCG method with variable preconditioning resembling the standard convergence rate bound of the PCG method with fixed preconditioning. Such stronger assumptions hold in many presently known real life applications of the PCG methods with variable preconditioning, but often require extra computational work, e.g., more inner iterations in the inner-outer iterative methods.

References.

[1] O. Axelsson and P. S. Vassilevski. Variable-step multilevel preconditioning methods. I. Selfadjoint and positive definite elliptic problems. *Numer. Linear Algebra Appl.*, 1(1):75–101, 1994.

[2] O. Axelsson and P. S. Vassilevski. A black box generalized conjugate gradient solver with inner iterations and variable-step preconditioning. *SIAM J. Matrix Anal. Appl.*, 12(4):625–644, 1991.

[3] O. Axelsson. *Iterative solution methods*. Cambridge University Press, Cambridge, 1994.

[4] E. G. D′yakonov. *Optimization in solving elliptic problems*. CRC Press, Boca Raton, FL, 1996.

[5] G. E. Forsythe. On the asymptotic directions of the s-dimensional optimum gradient method. *Numer. Math.*, 11:57–76, 1968.

[6] G. H. Golub and Q. Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM J. Sci. Comput.*, 21(4):1305–1320, 1999/00.

[7] L. V. Kantorovič. On the method of steepest descent. *Doklady Akad. Nauk SSSR (N. S.)*, 56:233–236, 1947.

[8] L. V. Kantorovich and G. P. Akilov. *Functional Analysis in Normed Spaces*. Pergamon, NY, 1964.

[9] A. V. Knyazev and I. Lashuk. Steepest descent and conjugate gradient methods with variable preconditioning. Electronic. math.NA/0605767, arXiv.org, http://arxiv.org/abs/math/0605767, 2006-2007.

[10] K. Neymeyr. A geometric theory for preconditioned inverse iteration. I. Extrema of the Rayleigh quotient. *Linear Algebra Appl.*, 322(1-3):61–85, 2001.

[11] Y. Notay. Flexible conjugate gradients. *SIAM J. Sci. Comput.*, 22(4):1444–1460, 2000.

[12] V. Simoncini and D. B. Szyld. Flexible inner-outer Krylov subspace methods. *SIAM J. Numer. Anal.*, 40(6):2219–2239, 2003.

[13] V. Simoncini and D. B. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM J. Sci. Comput.*, 25(2):454–477, 2003.

[14] V. Simoncini and D. B. Szyld. On the occurrence of superlinear convergence of exact and inexact Krylov subspace methods. *SIAM Rev.*, 47(2):247–272, 2005.